
UNIVERSITI SAINS MALAYSIA

First Semester Examination
2010/2011 Academic Session

November 2010

MSG 387 – Computer Graphics
[Grafik Komputer]

Duration : 3 hours
[Masa : 3 jam]

Please check that this examination paper consists of ELEVEN pages of printed material before you begin the examination.

[Sila pastikan bahawa kertas peperiksaan ini mengandungi SEBELAS muka surat yang bercetak sebelum anda memulakan peperiksaan ini.]

Instructions : Answer **all six** [6] questions.

Arahan : Jawab **semua enam** [6] soalan.]

The question papers **shall not be taken out** from the examination hall and will be collected by invigilators.

Kertas soalan ini tidak boleh dibawa keluar daripada dewan peperiksaan dan akan dikutip oleh pengawas peperiksaan.

In the event of any discrepancies, the English version shall be used.

[Sekiranya terdapat sebarang percanggahan pada soalan peperiksaan, versi Bahasa Inggeris hendaklah diguna pakai].

- 1) A raster system is designed using a 10-inch \times 10-inch screen with a resolution of 640 \times 480 pixels, and a pixel depth of 3 bytes per pixel. The refresh rate is 60Hz.
- What size of the frame buffer (in bytes) is needed?
 - What is the aspect ratio of the screen?
 - What is the transfer rate in KB (kilobytes) per second?
 - Given the horizontal retrace time is 5 microseconds (or 5.0×10^{-6} seconds) and vertical retrace time is 500 microseconds (or 500×10^{-6} seconds). What is the fraction of the total time spent in retrace of the electron beam to the total refresh time per frame?
 - How many distinct color choices (intensity levels) are available?
 - What is the maximum different colors we could display at any one time?

[10 marks]

- 2) Explain the following terms. You may illustrate your answers with the assistance of figures.
- Raster displays
 - Pixel
 - Frame Buffer
 - Scan conversion
 - Vertical retrace of a raster monitor

[10 marks]

- 3) Answer the questions below. You may illustrate your answers with the assistance of figures.
- What is line cap?
 - Explain 3 types of line caps in detail.
 - When two thick lines are joined with each other, they create an angle between them. This joint can be made smooth by using one of the 3 joining methods, i.e. miter joint, round joint or bevel joint. Explain these 3 joints in detail.

[20 marks]

- 1) Suatu sistem raster direka dengan menggunakan skrin bersaiz 10-inci \square 10-inci, berleraian 640 \square 480 piksel, dan kedalaman piksel 3 bait per piksel. Kadar segar semula ialah 60Hz.
- Apakah saiz kerangka penimbal (dalam ukuran bait) yang diperlukan?
 - Apakah nisbah bidang bagi skrin tersebut?
 - Apakah kadar pemindahan dalam KB (kilobait) per saat?
 - Diberi masa surih balik mendatar ialah 5 mikro-saat (atau $5.0 \square 10^{-6}$ saat) dan masa surih menegak ialah 500 mikro-saat (atau $5.0 \square 10^{-6}$ saat). Apakah pecahan bagi jumlah masa yang diambil dalam surih balik oleh pancaran elektron kepada jumlah masa segar semula bagi satu kerangka?
 - Berapakah pilihan warna yang berasingan (peringkat keamatan) yang boleh didapati?
 - Apakah maksima warna yang berbeza yang dapat dipaparkan pada sesuatu masa yang sama?

[10 markah]

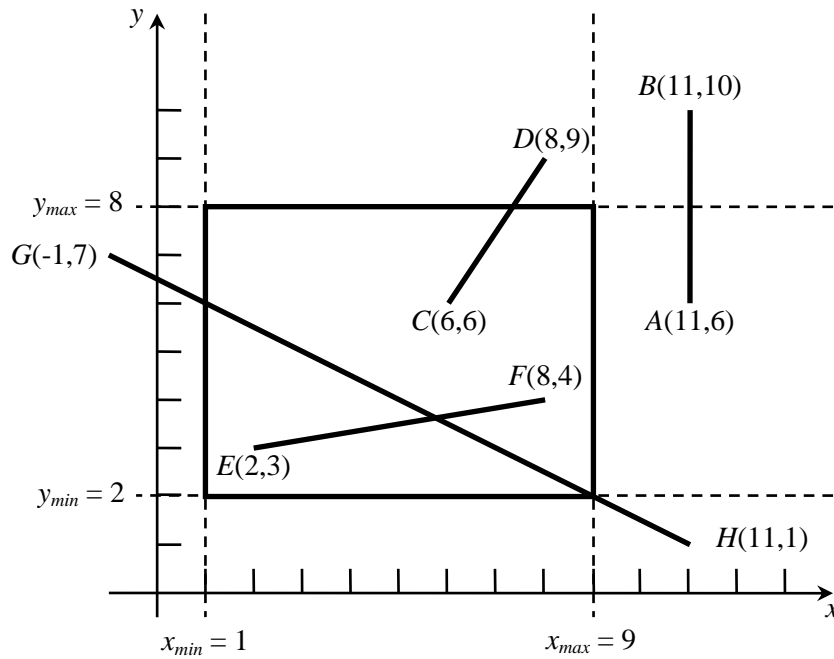
- 2) Jelaskan istilah-istilah di bawah. Anda boleh huraikan jawapan anda dengan bantuan gambarajah.
- Paparan raster
 - Piksel
 - Penimbal kerangka
 - Penukaran imbas
 - Surih balik menegak bagi suatu monitor raster

[10 markah]

- 3) Jawab soalan-soalan di bawah. Anda boleh huraikan jawapan anda dengan bantuan gambarajah.
- Apakah tukup garis?
 - Jelaskan 3 jenis tukup garis dengan terperinci.
 - Apabila dua garis tebal dicantum, satu sudut terwujud di antara kedua-dua garis tebal tersebut. Cantuman ini boleh dilicinkan dengan menggunakan salah satu daripada tiga kaedah cantuman, iaitu cantuman rencong, cantuman bulat atau cantuman serong. Jelaskan ketiga-tiga cantuman tersebut dengan mendalam.

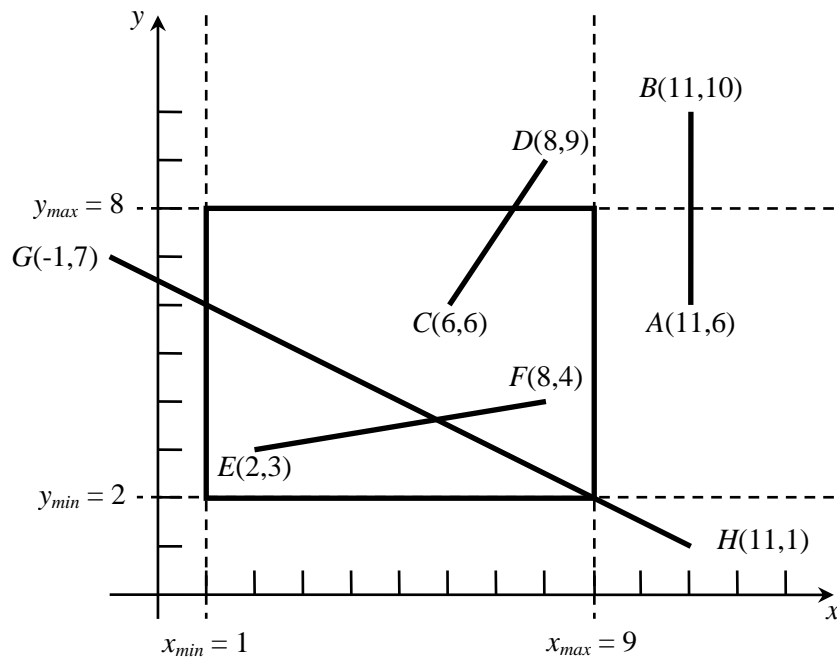
[20 markah]

- 4) Answer the questions below. You may illustrate your answers with the assistance of figures.
- Describe in brief Liang-Barsky Algorithm for two dimensional line clipping.
 - Given the figure below, the display window region is defined by x_{min} , x_{max} , y_{min} and y_{max} , use the Liang-Barsky Algorithm to clip the lines AB , CD , EF and GH .



[20 marks]

- 4) Jawab soalan-soalan di bawah. Anda boleh huraikan jawapan anda dengan bantuan gambarajah.
- Huraikan dengan ringkas Algoritma Liang-Barsky untuk potongan garis.
 - Diberi gambarajah di bawah, kawasan tettingkap paparan ditakrifkan oleh x_{min} , x_{max} , y_{min} dan y_{max} , gunakan Algoritma Liang-Barsky untuk memotong garis-garis AB, CD, EF dan GH.



[20 markah]

Refer to C++/OpenGL program below to answer question 5,

```
#include <stdlib.h>
#include <GL/glut.h>

// rasters is a bitmap array which has 32x32 bits in width and height
GLubyte rasters[] = {
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C,

    0xF8, 0x3F, 0xF0, 0x7C, 0xF8, 0x3F, 0xF0, 0x7C,
    0xF8, 0x3F, 0xF0, 0x7C, 0xF8, 0x3F, 0xF0, 0x7C,
    0xF8, 0x3F, 0xF0, 0x7C,

    0xFF, 0xC0, 0x0F, 0xFC, 0xFF, 0xC0, 0x0F, 0xFC,
    0xFF, 0xC0, 0x0F, 0xFC, 0xFF, 0xC0, 0x0F, 0xFC,
    0xFF, 0xC0, 0x0F, 0xFC,

    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C,

    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};

void init(void)
{
    glPixelStorei (GL_UNPACK_ALIGNMENT, 4);
    glClearColor (1.0, 1.0, 1.0, 1.0);
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f (1.0, 0.0, 0.0);
    glRasterPos2i (10, 10);
    glBitmap (30, 30, 0.0, 0.0, 40.0, 0.0, rasters);
    glColor3f (0.0, 1.0, 0.0);
    glBitmap (30, 30, 0.0, 0.0, 0.0, 0.0, rasters);
    glColor3f (0.0, 0.0, 1.0);
    glBitmap (30, 30, -40.0, 0.0, 40.0, 0.0, rasters);
    glFlush();
}

void reshape(int w, int h)
{
    glViewport(0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D (0, w, 0, h);
    glMatrixMode(GL_MODELVIEW);
}
```

Rujuk kepada program C++/OpenGL di bawah untuk menjawab soalan 5,

```
#include <stdlib.h>
#include <GL/glut.h>

// rasters is a bitmap array which has 32x32 bits in width and height
GLubyte rasters[] = {
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C,

    0xF8, 0x3F, 0xF0, 0x7C, 0xF8, 0x3F, 0xF0, 0x7C,
    0xF8, 0x3F, 0xF0, 0x7C, 0xF8, 0x3F, 0xF0, 0x7C,
    0xF8, 0x3F, 0xF0, 0x7C,

    0xFF, 0xC0, 0x0F, 0xFC, 0xFF, 0xC0, 0x0F, 0xFC,
    0xFF, 0xC0, 0x0F, 0xFC, 0xFF, 0xC0, 0x0F, 0xFC,
    0xFF, 0xC0, 0x0F, 0xFC,

    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C, 0xF8, 0x00, 0x00, 0x7C,
    0xF8, 0x00, 0x00, 0x7C,

    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};

void init(void)
{
    glPixelStorei (GL_UNPACK_ALIGNMENT, 4);
    glClearColor (1.0, 1.0, 1.0, 1.0);
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f (1.0, 0.0, 0.0);
    glRasterPos2i (10, 10);
    glBitmap (30, 30, 0.0, 0.0, 40.0, 0.0, rasters);
    glColor3f (0.0, 1.0, 0.0);
    glBitmap (30, 30, 0.0, 0.0, 0.0, 0.0, rasters);
    glColor3f (0.0, 0.0, 1.0);
    glBitmap (30, 30, -40.0, 0.0, 40.0, 0.0, rasters);
    glFlush();
}

void reshape(int w, int h)
{
    glViewport(0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D (0, w, 0, h);
    glMatrixMode (GL_MODELVIEW);
}
```

```

void keyboard(unsigned char key, int x, int y)
{
    switch (key) {
        case 27: //Esc
            exit(0);
    }
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(150, 100);
    glutInitWindowPosition(10, 10);
    glutCreateWindow("MSG387 Exam");
    init();
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

The syntax of `glBitmap` function is given as below,

```

void glBitmap (
    GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig,
    GLfloat xmove, GLfloat ymove, const GLubyte *bitmap )

```

Parameters

<i>width, height</i>	Specify the pixel width and height of the bitmap image.
<i>xorig, yorig</i>	Specify the location of the origin in the bitmap image. The origin is measured from the lower left corner of the bitmap, with right and up being the positive axes.
<i>xmove, ymove</i>	Specify the <i>x</i> and <i>y</i> offsets to be added to the current raster position after the bitmap is drawn.
<i>bitmap</i>	Specifies the address of the bitmap image.


```

void keyboard(unsigned char key, int x, int y)
{
    switch (key) {
        case 27: //Esc
            exit(0);
    }
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(150, 100);
    glutInitWindowPosition(10, 10);
    glutCreateWindow("MSG387 Exam");
    init();
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

Sintaks bagi fungsi glBitmap diberi seperti di bawah,

```

void glBitmap (
    GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig,
    GLfloat xmove, GLfloat ymove, const GLubyte *bitmap )

```

Parameter-parameter

<code>width, height</code>	<i>Nyatakan lebar dan tinggi dalam ukuran piksel bagi imej bitmap.</i>
<code>xorig, yorig</code>	<i>Nyatakan lokasi untuk asalan imej bitmap. Asalan tersebut diukur dari bawah pepenjuru kiri bitmap, dengan kanan dan tegak sebagai paksi-paksi positif.</i>
<code>xmove, ymove</code>	<i>Nyatakan offset-offset x dan y yang akan ditambah kepada kedudukan raster semasa setelah bitmap dilukis.</i>
<code>bitmap</code>	<i>Nyatakan alamat imej bitmap.</i>

- 5) Answer the following questions by referring to the program in pages 6 and 8.
- What does “callback function” mean? Use an example in your explanation.
 - In the program, what does the function `keyboard` do? Explain the parameters used in the `keyboard` function.
 - How does `glViewport` affect the display of the pictures on the display window? Explain by using an example.
 - What will happen when we resize the display window if `glutReshapeFunc` does not exist in the program? Explain by using an example.
 - `rasters` is a bitmap array which has 32×32 bits in width and height. Draw the output of this program on the graph paper provided, assuming that the units in the graph paper are in pixels, and the screen resolution is 230×180 , i.e. the same size as the graph paper.

[20 marks]

- 6) Given a polygon as defined in the function `draw_triangle` below,

```
void draw_triangle(void)
{
    glBegin (GL_LINE_LOOP);
        glVertex2f(10.0, 5.0);
        glVertex2f(40.0, 30.0);
        glVertex2f(20.0, 30.0);
    glEnd();
}
```

On the graph paper provided, draw the polygon after each transformation as stated in the program segment below.

```
:
glColor3f (0.0, 0.0, 0.0);
glLineWidth(2);
glLoadIdentity ();
draw_triangle ();

glRotatef (90.0, 0.0, 0.0, 1.0);
glLineWidth(1);
glColor3f (1.0, 0.0, 0.0);
draw_triangle ();

glScalef (-2.0, 1.0, 1.0);
glColor3f (0.0, 1.0, 0.0);
draw_triangle ();

glTranslatef (-10.0, -30.0, 0.0);
glColor3f (0.0, 0.0, 1.0);
draw_triangle ();

glLoadIdentity ();
glTranslatef (20.0, -10.0, 0.0);
glColor3f (1.0, 1.0, 0.0); //yellow color
draw_triangle ();
:
```

[20 marks]

...11/-

- 5) Jawab soalan-soalan di bawah dengan merujuk kepada program di mukasurat 7 dan 9.
- Apakah maksud “fungsi panggil balik”? Gunakan contoh dalam penjelasan anda.
 - Apakah tujuan fungsi keyboard dalam program di atas? Jelaskan parameter-parameter yang digunakan oleh fungsi keyboard.
 - Bagaimanakah `glViewport` mempengaruhi paparan gambar atas tettingkap paparan? Jelaskan dengan menggunakan satu contoh.
 - Apakah yang akan berlaku bila kita mengubah saiz tettingkap paparan jika `glutReshapeFunc` tidak wujud di dalam program tersebut? Jelaskan dengan menggunakan satu contoh.
 - `rasters` ialah satu tatasusunan bitmap yang mempunyai 32×32 bit lebar dan tinggi. Lukiskan output bagi program ini di atas kertas graf yang disediakan, anggapkan unit-unit di atas kertas graf adalah dalam ukuran piksel, dan leraian skrin ialah 230×180 , iaitu bersaiz sama dengan kertas graf.

[20 markah]

- 6) Diberi suatu poligon seperti yang didefinisikan dalam fungsi `draw_triangle` di bawah,

```
void draw_triangle(void)
{
    glBegin (GL_LINE_LOOP);
        glVertex2f(10.0, 5.0);
        glVertex2f(40.0, 30.0);
        glVertex2f(20.0, 30.0);
    glEnd();
}
```

Di atas kertas graf yang disediakan, lukiskan polygon selepas setiap transformasi seperti yang dinyatakan dalam segmen program di bawah.

```
:
glColor3f (0.0, 0.0, 0.0);
glLineWidth(2);
glLoadIdentity ();
draw_triangle ();

glRotatef (90.0, 0.0, 0.0, 1.0);
glLineWidth(1);
glColor3f (1.0, 0.0, 0.0);
draw_triangle ();

glScalef (-2.0, 1.0, 1.0);
glColor3f (0.0, 1.0, 0.0);
draw_triangle ();

glTranslatef (-10.0, -30.0, 0.0);
glColor3f (0.0, 0.0, 1.0);
draw_triangle ();

glLoadIdentity ();
glTranslatef (20.0, -10.0, 0.0);
glColor3f (1.0, 1.0, 0.0); //yellow color
draw_triangle ();
:
```

[20 markah]